

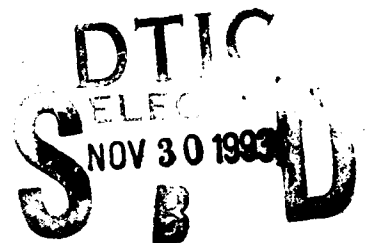
AD-A273 287



SSI-TR-28

GETTING THE GOOD BOUNCE:  
TECHNIQUES FOR EFFICIENT  
MONTE CARLO ANALYSIS OF  
COMPLEX REACTING FLOWS

1983



Prepared by

James B. Elgin

Spectral Sciences, Inc.  
99 So. Bedford Street  
Burlington, MA 01803

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

93-29111



93 11 26 133

## TABLE OF CONTENTS

SECTION	PAGE
ABSTRACT .....	1
1. OVERVIEW OF THE DIRECT SIMULATION MONTE CARLO METHOD .....	2
2. GAS MODEL AND EQUILIBRIUM PROPERTIES .....	4
2.1 Preliminary Equilibrium Gas Relations .	4
2.2 Analytical Form of the Collision Cross Section .....	5
2.3 Equilibrium Reference Properties for a Multi-Component Gas .....	8
2.4 Internal Energy Model .....	10
3. INTERNAL REPRESENTATION .....	11
3.1 State Vector .....	11
3.2 Reduction to a Reasonable Number of Simulated Molecules .....	12
3.3 Internal Scales .....	12
3.4 Weighting Factors .....	14
4. COLLISION MECHANICS .....	16
4.1 Relations for Elastic Collisions .....	16
4.2 Simulation of Inelastic Collisions ...	18
4.3 Collisions for Molecules with Distinct Weighting Factors .....	21

## TABLE OF CONTENTS (CONT.)

SECTION	PAGE
4.4 Reactive Collisions .....	22
4.5 Post-Reaction State Definition .....	24
4.6 Dissociative Reactions .....	25
5. MOLECULAR TRANSLATIONS .....	27
5.1 Translation in an Axisymmetric Coordinate System .....	27
5.2 Molecular Cloning .....	29
6. COLLISION SAMPLING IN A MULTI-COMPONENT VHS GAS .....	32
6.1 General Considerations and Approach ..	32
6.2 Collision Sampling for a Single Component Gas .....	32
6.2.1 Collision Pair Selection .....	33
6.2.2 Collision Time Counter for a Single Component Gas .....	34
6.3 Collision Class Sampling in Gas Mixtures .....	37
6.4 Global Collision Sampling in a Gas Mixture .....	38
6.4.1 Global Collision Time Counter .	38
6.4.2 Collision Pair Selection in Multi-Component Mixtures .....	41
6.4.3 Summary of Collision Sampling in Multi-Component Mixtures ...	41

## TABLE OF CONTENTS (CONT.)

SECTION		PAGE
7.	INITIAL AND BOUNDARY CONDITIONS .....	43
7.1	General Considerations .....	43
7.2	Initial Conditions .....	43
7.2.1	Number of Simulated Molecules and Weighting Factors .....	44
7.2.2	Initial Positions .....	45
7.2.3	Initial Velocity Components ...	45
7.2.4	Initial Internal Energies .....	46
7.3	Wall Boundary Conditions .....	48
7.4	Free Gas Boundary Conditions .....	49
7.4.1	Incoming Number Flux .....	49
7.4.2	Incoming Molecular Velocity Components .....	50
7.4.3	Incoming Molecular Position ...	51
7.4.4	Incoming Molecular Internal Energies .....	51
8.	STATISTICAL SAMPLING OF OUTPUT .....	52
8.1	General Considerations .....	52
8.2	Sampling of Instantaneous Output Quantities .....	53
8.3	Sampling of Time Averaged Quantities .	56
9.	REFERENCES .....	58

# GETTING THE GOOD BOUNCE: TECHNIQUES FOR EFFICIENT MONTE CARLO ANALYSIS OF COMPLEX REACTING FLOWS

## ABSTRACT

The direct simulation Monte Carlo method as it is applied to complex reacting flows is discussed at length. Relations are presented for the efficient treatment of energy dependent cross sections, multiple species and chemical reactions in multi-dimensional flows. Special emphasis is placed on recent advancements in the method developed at Spectral Sciences, including the use of dynamically adjustable weighting factors and global time counters.

DTIC QUALITY INSPECTED 5

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability; Codes	
Dist	Avail and/or Special
A-1	

## 1. OVERVIEW OF THE DIRECT SIMULATION MONTE CARLO METHOD

The direct simulation Monte Carlo method, as pioneered by G. A. Bird,<sup>(1)</sup> provides a powerful technique for the simulation of real gas flows. It bridges the gap between continuum and free molecular flow, retaining validity in either extreme. It can be used to describe complex mixtures, including effects of chemical reactions, heat conduction, viscosity and diffusion for flows in three dimensions. To date, it is the only approach which can demonstrate these abilities for general flow configurations.

The basic calculational technique is well described by its originator in Ref. 1, to which frequent reference will be made. The present purpose is to describe how the technique is implemented at Spectral Sciences, Inc. (SSI), with special emphasis on extensions developed at SSI and elsewhere after the publication of Ref. 1. Elementary concepts and relations which are essential to a coherent explanation are included here for clarity.

The direct simulation Monte Carlo method involves storing a discrete number of molecules (via their velocities, positions and other pertinent information) in a computer. The solution region is broken up into a number of separate cells, and the solution is stepped forward in time in a two stage process. First, the molecules are advanced along their trajectories by an amount appropriate to their velocity and a time increment  $\Delta t_m$ . In this first stage some molecules will leave the solution region and some will be introduced as determined by the boundary conditions for a particular problem. The second stage is to

simulate collisions in each cell appropriate to  $\Delta t_m$  so that collision frequencies are properly simulated. A basic hypothesis of the method is that if the time step is made small enough the processes of translations and collisions can be uncoupled in this manner.

Periodically, the solution is sampled by accumulating statistical sums of number densities, velocities and other basic properties. The solution is run repeatedly until statistical deviations are reduced to a desired limit, and then physically meaningful output quantities are computed from the statistical sums. The number of molecules represented is typically a few thousand at a time, which is vastly fewer than the number of molecules occurring in virtually all real flows. Hence, the construction of a dynamically similar flow to be simulated in the computer is an essential feature of the method.

## 2. GAS MODEL AND EQUILIBRIUM PROPERTIES

### 2.1 Preliminary Equilibrium Gas Relations

For most problems of interest there is a far field equilibrium state whose properties are of relevance to the problem to be solved. Frequently length and velocity scales for the problem are obtained from the far field state and used to nondimensionalize the internal code variables. Even when the far field state is not used for scaling purposes, it still provides an important comparison case.

For a rest gas in equilibrium the normalized distribution function for the relative speed,  $c_r$ , between molecules of species  $i$  and molecules of species  $j$  is given by<sup>(2)</sup>

$$f_{ij}(c_r) = \frac{4a_{ij}^{3/2}}{\sqrt{\pi}} c_r^2 \exp(-a_{ij}c_r^2) \quad , \quad (1)$$

where

$$a_{ij} = \frac{\mu_{ij}}{2R_0T_\infty} \quad , \quad (2)$$

and  $\mu_{ij}$  is the reduced mass of the pair; i.e.

$$\mu_{ij} = \frac{m_i m_j}{m_i + m_j} \quad , \quad (3)$$

with  $m_i$  and  $m_j$  representing the masses of the two species. In these relations,  $T_\infty$  is the far field temperature and  $R_0$  is the universal gas constant. ( $R_0$  is used instead of Boltzmann's constant since the molecular masses will be consistently



represented in atomic mass units rather than grams.) The available translational collision energy between two molecules,  $E_c$ , is given by

$$E_c = \frac{1}{2} \mu_{ij} c_r^2 \quad (4)$$

## 2.2 Analytical Form of the Collision Cross Section

Whenever the direct simulation Monte Carlo method is applied, it is necessary to make tradeoffs between accuracy and simplicity in molecular models. It does no good to use a complex molecular potential surface and then find that reasonable computer run times result in very large statistical fluctuations in the output. Since the final output will reflect errors in the statistics as well as errors in the models, there is a strong impetus to use models which contain the essential physics, but which can be applied in a computationally efficient manner. The current state-of-the-art is the Variable-Hard-Sphere (VHS) model.<sup>(3)</sup> In this model molecules have a collision cross section which varies as an inverse power of the available collision energy. Hence, if  $\sigma_{ij}$  is the collision cross section for collisions of species  $i$  with species  $j$ , then  $\sigma_{ij}$  is given by a relation of the form

$$\sigma_{ij} = A_{ij} E_c^{-\omega} \quad , \quad (5)$$

where  $A_{ij}$  is a constant coefficient. It follows that the effective diameter for molecules of species  $i$ ,  $d_i$ , is implicitly defined as a function of available collision energy by the relation

$$\sigma_{ii} = \pi d_i^2 = A_{ii} E_c^{-\omega} \quad . \quad (6)$$

$A_{ii}$  can be determined from a reference cross section and velocity via

$$A_{ii} = \left[ \sigma_{ii} (m_i c_r^2 / 4)^\omega \right]_{\text{ref}} . \quad (7)$$

If a reference cross section is given for a reference temperature rather than a reference velocity, then the usual choice for the reference velocity is that velocity which has a collision energy equal to the mean collision energy occurring in collisions at the reference temperature. Mathematically, this is equivalent to

$$\left( c_r^2 \right)_{\text{ref}} = \frac{\overline{c_r^3 \sigma_{ii}}}{\overline{c_r \sigma_{ii}}} , \quad (8)$$

where the bars over the quantities indicate averages taken over the distribution function given in Eq. (1) evaluated for  $m_i = m_j$  and  $T_\infty = T_{\text{ref}}$ . Equation (8) can be simplified to give

$$\left( c_r^2 \right)_{\text{ref}} = \frac{4(2 - \omega) R_0 T_{\text{ref}}}{m_i} . \quad (9)$$

For simulations involving a large number of species, reference cross sections are frequently not available for all possible collision pairs. In this case it is possible to specify  $A_{ii}$  for self collisions only, and then use Eq. (6) to get a molecular diameter as a function of collision energy. Then, applying the relation

$$\sigma_{ij} = \pi \left[ (d_i + d_j) / 2 \right]^2 \quad (10)$$

the coefficient in Eq. (5) for interspecie collisions is given by

$$A_{ij} = \left[ \frac{1}{2} \left( \sqrt{A_{ii}} + \sqrt{A_{jj}} \right) \right]^2 . \quad (11)$$

For the internal workings of a Monte Carlo code, it is usually more convenient to express the collision cross section as a function of the relative collision velocity rather than the collision energy. This is simply achieved via the relation

$$\sigma_{ij} = B_{ij} c_r^{-2\omega} , \quad (12)$$

where

$$B_{ij} = A_{ij} \left( \frac{\mu_{ij}}{2} \right)^{-\omega} . \quad (13)$$

The parameter  $\omega$  can be related to  $\eta$ , the exponent of distance in an inverse power intermolecular force law via the relation<sup>(3)</sup>

$$\omega = 2/(\eta - 1) . \quad (14)$$

Hence, hard sphere molecules (for which  $\eta$  goes to infinity) are represented by  $\omega$  equal to zero. There is a substantial body of evidence, however, that the effective size of molecules does indeed decrease with increasing collision energy, so a positive value of  $\omega$  is usually a better choice.  $\omega$  can be determined from molecular beam data, or from its macroscopic implications. For example, if  $s$  is the temperature exponent for the coefficient of viscosity, then it can be shown<sup>(3)</sup> that

$$s = \omega + 1/2 , \quad (15)$$

so a measurement of the temperature dependence of the viscosity coefficient serves as an indirect determination of  $\omega$ .

In order to incorporate the model for internal energy transfer to be discussed in Section 4, it is necessary that  $\omega$  be assumed the same for all interactions. This represents one of the major restrictions in the current state of modeling.

Although the sizes of molecules are allowed to vary in the VHS molecular model in deciding whether or not a collision is to occur, when a collision does occur the post collision velocity components are computed as if it were a hard sphere collision (see Section 4). This results in a substantial computational simplification and yet retains good agreement with the macroscopic predictions of the more exact model.<sup>(3)</sup> (See Ref. 1 for a discussion of molecular scattering for general power law potentials.)

### 2.3 Equilibrium Reference Properties for a Multi-Component Gas

One advantage of the VHS model is that the molecules have a well defined cross section, so it is possible to define a mean free path without putting limitations on the minimum deflection angle that is considered. As is the general case for multi-component gases, however, each component has its own mean free path, and the overall mean free path for the mixture must be defined as a weighted average of the mean free paths of the individual species. The somewhat cumbersome relations required to calculate the overall mean free path are given here. It should be noted that the mean free path is calculated only once for a given problem, so the computational effort required to evaluate it is completely negligible.

An individual molecule of species  $i$  will suffer collisions with molecules of species  $j$  with a frequency  $\nu_i^j$  given by

$$\nu_i^j = n_{j\infty} \overline{\sigma_{ij} c_r} \quad (16)$$

where  $n_{j\infty}$  is the number density of species  $j$  and  $\overline{\sigma_{ij} c_r}$  is the average product of cross section times relative velocity for the two species, obtained by integrating over the distribution function given in Eq. (1). When this operation is performed, the result is

$$v_i^j = \frac{2B_{ij} n_{j\infty}}{\pi} \Gamma(2 - \omega) a_{ij}^{\omega - 1/2} , \quad (17)$$

where  $\Gamma$  denotes the gamma function.

The total collision frequency for an individual molecule of species  $i$ ,  $v_i$ , is obtained by summing Eq. (15) over all species, i.e.

$$v_i = \sum_{j=1}^p v_i^j \quad (18)$$

and the mean free path,  $\lambda_i$ , for molecules of species  $i$  is given by

$$\lambda_i = \bar{c}_i / v_i = \sqrt{\frac{8R_0 T_\infty}{\pi m_i}} / v_i , \quad (19)$$

where  $\bar{c}_i$  is the mean molecular speed for species  $i$  molecules. The mean free path for the gas mixture,  $\lambda_\infty$ , is then defined as the number density weighted average of the  $\lambda_i$  via

$$\lambda_\infty = \sum_{i=1}^p \frac{n_{i\infty} \lambda_i}{n_\infty} , \quad (20)$$

where  $n_\infty$  is the total number density:

$$n_\infty = \sum_{i=1}^p n_{i\infty} . \quad (21)$$

A useful velocity scale is given by  $v_s$ , defined by

$$v_s = \sqrt{\frac{2R_0 T_\infty}{\bar{m}}} , \quad (22)$$

where  $\bar{m}$  is the reference mean molecular weight, i.e.

$$\bar{m} = \sum_{i=1}^P \frac{n_{i\infty} m_i}{n_{\infty}} \quad . \quad (23)$$

$v_s$  is the most probable molecular speed for molecules of the mean molecular weight at the reference temperature.

#### 2.4 Internal Energy Model

The current state of modeling for internal energy effects in Monte Carlo flow field simulations is the phenomenological model of Borgnakke and Larsen.<sup>(4)</sup> In this model, transfer of energy between internal and translational modes is allowed, but it is necessary to assume that each species has a fixed number of internal degrees of freedom,  $\zeta_i$ . This is equivalent to assuming a constant specific heat,  $C_{pi}$ , for each species which can be related to the number of internal degrees of freedom via

$$\zeta_i = 2 \frac{C_{pi} m_i}{R_0} - 5 \quad . \quad (24)$$

Alternatively,  $\zeta_i$  can be related to the ratio of specific heats for species  $i$ ,  $\gamma_i$ , by the relation

$$\zeta_i = \frac{5 - 3\gamma_i}{\gamma_i - 1} \quad . \quad (25)$$

The interchange of internal and translational energy will be discussed in Section 4, and the selection of initial conditions will be considered in Section 7.

### 3. INTERNAL REPRESENTATION

#### 3.1 State Vector

Each simulated molecule in the direct simulation Monte Carlo method is represented by a state vector which comprises all of the information the code has with regard to that particular molecule. The state vector has:

- Position element(s) defining the location of the molecule in the coordinate system being used. For axisymmetric simulations, this is a radial and an axial element.
- Three velocity elements. A molecular collision is always considered as a three dimensional event, regardless of the overall dimensionality of the problem. For spatially one dimensional problems it is possible to store only two pieces of velocity information and compute the required three velocity components as needed for collision sampling. This is an example of the frequent tradeoff which must be made between storage and computing requirements.
- A value for the internal energy of the molecule. Note that the basic model does not discriminate between internal modes for a particular species. This can be done, if desired, by introducing separate species for the separate modes.
- An indicator determining the molecular species. This indicator in turn implies all of the properties associated with that species (molecular weight, number of internal degrees of freedom, name, etc.).

- An indicator giving the cell in which the molecule currently resides. It is possible to avoid allocating this particular storage element, but it usually results in enough computational simplification to justify its use.

### 3.2 Reduction to a Reasonable Number of Simulated Molecules

It is clearly impossible to run a computer simulation with anywhere near the same number of molecules that exist in the actual flow problem. The adjustment that is made to make the simulation possible is to artificially increase the cross section, and decrease the number density, by a large factor. It is the product of number density and cross section which determines the collision frequency for a given molecule, and it is the collision frequency which must be correctly simulated if the correspondence between the real and simulated flows is to be correct. This is an essential feature of the direct simulation method which has not always been adequately emphasized. It means that the internal scaling factors do not proceed on a strictly dimensional basis. For example, the scaling factor for cross sections is not the square of the scaling factor for lengths.

### 3.3 Internal Scales

Many problems are more reasonably handled if the internal calculations are carried out with scaled or dimensionless values. This avoids possible problems such as numerical overflow which can cause an execution time error. Such errors can be particularly insidious and difficult to locate in a code whose very essence involves the random combination of numbers. Furthermore,



examination of scaled values makes the detection of erroneous values easier while debugging codes since large exponents are usually indicative of an error when the variables are internally scaled. At SSI, at least, the output is produced in physically meaningful dimensional form. Hence, the scaling that is discussed here is irrelevant (or nearly so) to the interpretation of code output; it is strictly a matter of the internal representation.

The obvious choices for length and velocity scales are  $\lambda_\infty$  and  $v_s$  as defined in Section 2, which are used to nondimensionalize the position and velocity elements of the state vector. There is no need to provide further nondimensionalization of mass beyond representing them in atomic mass units, so none is provided. Hence, the scaling factor for energy is just  $v_s^2$ , which is used to nondimensionalize the internal energy element of the state vector.

Number densities are scaled with respect to the far field reference number density,  $n_\infty$ , which leaves only the cross section scaling factor to be determined. This factor follows from the condition of flow similarity, which requires that the probability of a molecule suffering a collision in traveling a given path length be accurately simulated. This dimensionless probability can be expressed as the product of a cross section times a number density times a path length (at least for small enough path lengths), and it is required that this product be the same for dimensional and scaled representations. This implies that the product of the scaling factors for these three quantities be unity and, therefore, the cross section scaling factor is  $1/(n_\infty \lambda_\infty)$ . The internal scaling factors used for the SSI Monte Carlo codes are summarized in Table 1.

TABLE 1 - SCALING FACTORS USED FOR THE INTERNAL  
REPRESENTATION OF QUANTITIES IN THE SSI  
DIRECT SIMULATION MONTE CARLO CODES.  
ALL VARIABLES ARE DEFINED IN SECTION 2.

PROPERTY	SCALING FACTOR
Length	$\lambda_{\infty}$
Velocity	$v_s$
Time	$\lambda_{\infty}/v_s$
Number Density	$n_{\infty}$
Mass	a.m.u.
Energy	$(\text{a.m.u.})v_s^2$
Cross Section	$1/(n_{\infty}\lambda_{\infty})$

### 3.4 Weighting Factors

Weighting factors are a crucial element of a successful Monte Carlo simulation, allowing trace species to be described with reasonable statistics. The weighting factor is the number of "real" molecules that corresponds to each "simulated" molecule. A "simulated" molecule corresponds to one molecule's worth of storage (one state vector) allocated in the computer, and the weighting factor is its statistical weight. So, for example, the total number density in a cell might be represented

$$n_{\text{cell}} = \sum_{i=1}^P \frac{N_i W_i}{V} \quad , \quad (26)$$

where  $N_i$  indicates the number of simulated molecules of species  $i$  in the cell,  $W_i$  is the weighting factor for that species in that cell,  $V$  is the cell volume and  $p$  is the number of species. The product  $N_i W_i$  that appears in Eq. (26) is termed the number of "real" molecules of species  $i$  in the cell. Note that  $n_{\text{cell}}$  as calculated by Eq. (26) is a scaled value; it would have to be multiplied by  $n_{\infty}$ , as shown in Table 1, to become a dimensional evaluation of the number density.

The weighting factors used in the SSI codes are dependent on cell and species. Hence, flowfields where a given species is much more dominant in one portion of the solution region than another can be accurately represented. It is possible, of course, to make weighting factors functions of other variables, such as velocity, for specialized purposes.

A critical error that can occur in Monte Carlo codes is to have the number of simulated molecules exceed the dimensioned limit of the code. On the other hand, it is generally desirable to have as many molecules as is feasible to obtain good statistics. Resolution of these conflicting considerations is complicated by lack of a priori knowledge of what the species number densities will be as a function of space and time. The way the resolution is achieved in the SSI codes is by a dynamic adjustment of the weighting factors, as required. This keeps the number of simulated molecules more or less constant while allowing the number of real molecules to adjust as the solution evolves. The introduction of weighting factors, with the ability to adjust them as the solution demands, is an important feature of a successful Monte Carlo description.

## 4. COLLISION MECHANICS

### 4.1 Relations for Elastic Collisions

The purpose of this section is to present relations appropriate to the simulation of a collision in a Monte Carlo flowfield code. (The question of how molecules are selected for collisions, which is crucial to the proper simulation of collision frequency, will be taken up in Section 6.) Conservation of momentum implies that the center-of-mass velocity of the collision pair is unchanged by the collision; and conservation of energy then implies that the magnitude of the relative velocity between the collision partners is also unchanged by the collision.<sup>(5)</sup> Since the collision is treated as a statistical event, all that remains is to select the direction of the post-collision relative velocity vector from the correct distribution. As mentioned in Section 2.2, collisions in the VHS model are treated as hard sphere collisions when they occur (though they do not occur with the same velocity dependence as do hard sphere collisions). Hence, as far as the collision mechanics is concerned, the model is a hard sphere model. For hard sphere molecules, all directions for the post-collision relative velocity vector are equally likely. This is the chief computational simplicity of the VHS model.

Let the two molecules be identified by subscripts 1 and 2, with  $m$  and  $\vec{v}$  denoting their masses and velocities. If  $i$  and  $f$  indicate initial and final states, then the relations for the collision can be summarized via:

$$\vec{v}_{cm} = \frac{m_1 \vec{v}_{1i} + m_2 \vec{v}_{2i}}{m_1 + m_2} , \quad (27)$$

$$v_r = |\vec{v}_{1i} - \vec{v}_{2i}| , \quad (28)$$

$$\cos(\theta) = 1 - 2\beta , \quad (29)$$

$$\sin(\theta) = \sqrt{1 - \cos^2(\theta)} , \quad (30)$$

$$\phi = 2\pi\beta , \quad (31)$$

$$\vec{v}_{rf} = v_r \left[ \cos(\theta), \sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi) \right] , \quad (32)$$

$$\vec{v}_{1f} = \vec{v}_{cm} + \frac{m_2}{m_1 + m_2} \vec{v}_{rf} , \quad (33)$$

and

$$\vec{v}_{2f} = \vec{v}_{cm} - \frac{m_1}{m_1 + m_2} \vec{v}_{rf} . \quad (34)$$

In these relations, and throughout this report,  $\beta$  indicates a random variable which is evenly distributed on the interval zero to one. Each time that  $\beta$  appears a different evaluation of the random variable is implied. Note that the expression for the post-collision relative velocity vector (Eq. (32)) is not coordinate system specific. The indicated vector components can apply to any locally orthogonal coordinate system, since the direction implied is random. The convenient coordinate system to use, of course, is the coordinate system used to define the velocity elements of the state vector. For axisymmetric simulations this will be radial, azimuthal and axial velocity components.

## 4.2 Simulation of Inelastic Collisions

The SSI codes use the Borgnakke and Larsen<sup>(4)</sup> phenomenological model for transfer of energy between internal and translational modes. In this model, a collision is assumed to be either perfectly elastic or perfectly inelastic, via a user specified probability. A perfectly inelastic collision is achieved by summing the total pre-collision energy (internal energy of both molecules plus the translational energy of their relative motion, Eq. (4)) and then assigning post-collision values from the equilibrium distribution for collisions with that total amount of energy, taking into account the number of internal degrees of freedom in the two molecules. Note that this model has the ability to relax from a nonequilibrium to an equilibrium state via an effective collision number. The ability to exchange internal energy in such a manner comprises a significant increase in capability for Monte Carlo codes beyond the previous models where molecules had no internal energy. It is this capability which enables the codes to realistically predict the macroscopic effects of polyatomic gas flow.

Let  $\zeta_1$  and  $\zeta_2$  be the number of internal degrees of freedom of the two molecules in an inelastic collision, and  $E_s$  be the total collision energy defined by

$$E_s = E_{ci} + E_{1i} + E_{2i} \quad , \quad (35)$$

where  $E_{ci}$  is the initial translational collision energy defined by Eq. (4), and  $E_{1i}$  and  $E_{2i}$  are the pre-collision internal energies of the two molecules. If  $\xi$  is defined by

$$\xi = E_{cf}/E_s \quad , \quad (36)$$

where  $E_{cf}$  is the post-collision translational energy, then  $\xi$  is selected according to the distribution

$$f(\xi) = A \xi^{1-\omega} (1-\xi)^b \quad (37)$$

where

$$A = \left[ \frac{\langle \zeta \rangle - \omega}{1 - \omega} \right]^{1-\omega} \left[ \frac{\langle \zeta \rangle - \omega}{\langle \zeta \rangle - 1} \right]^{\langle \zeta \rangle - 1}, \quad (38)$$

$$b = \langle \zeta \rangle - 1, \quad (39)$$

and

$$\langle \zeta \rangle = (\zeta_1 + \zeta_2)/2. \quad (40)$$

The sampling of Eq. (37) is done via a technique that is used frequently in Monte Carlo flowfield codes, the acceptance-rejection method. Equation (37) has been normalized so that the maximum value of the function is unity, and the parameter of the distribution,  $\xi$ , varies from zero to one. The sampling is done as follows:

- Choose a random value of  $\xi$ . I.e., set  $\xi$  equal to  $\beta$ , a random variable.
- Evaluate the distribution function for this value of  $\xi$ , and call it  $f_{\text{test}}$ .
- Get a second random variable, and check to see if it is greater or less than  $f_{\text{test}}$ . If it is greater than  $f_{\text{test}}$ , go back to the first step and repeat the process. If it is less than  $f_{\text{test}}$ , then keep the value of  $\xi$  obtained in the first step.

Note that the probability that any original value of  $\xi$  will be kept is proportional to  $f(\xi)$ . This is an extremely general technique and, as such, is very powerful. It is not always efficient, however, and direct sampling of distributions is usually to be

preferred if it can be accomplished. In this case, if one molecule is monatomic and the other is diatomic, i.e. if  $\langle \zeta \rangle = 1$ , then a direct sampling of the above distribution is achieved via

$$\xi = \beta^{[1/(2-\omega)]} , \quad (\langle \zeta \rangle = 1) . \quad (41)$$

Once  $\xi$ , and therefore the post-collision translation energy, is determined, the magnitude of the post-collision relative velocity is defined via

$$v_r = \sqrt{2E_{cf}/\mu_{12}} . \quad (42)$$

For inelastic collisions, this relation takes the place of Eq. (28) in the determination of the post-collision velocity elements of the state vectors.

The remainder of the collision energy must be divided up between the internal modes of the two molecules. If one of the molecules is monatomic (i.e., has zero internal degrees of freedom), then all of the internal energy goes to the other by default. Otherwise, if  $x$  is defined by

$$x = E_{1f}/(E_s - E_{cf}) , \quad (43)$$

where  $E_{1f}$  is the post-collision internal energy of the first molecule, then  $x$  is just the fraction of the total post-collision internal energy that ends up in the first molecule.  $x$  is distributed with a probability proportional to  $f(x)$ , given by

$$f(x) = B x^c (1-x)^d , \quad (44)$$

where

$$B = [(\langle \zeta \rangle - 2)^{(\langle \zeta \rangle - 2)}] / [c^c d^d] , \quad (45)$$

$$c = \zeta_1/2 - 1 , \quad (46)$$



and

$$d = \zeta_2/2 - 1 \quad . \quad (47)$$

Equation (44) can be sampled via the acceptance-rejection method to determine the allocation of the internal energy for the general case. For the special case of both molecules being diatomic, Eq. (44) becomes singular, with the limit being the trivial case that all distributions of internal energy are equally likely, i.e.

$$x = \beta \quad , \quad (\zeta_1 = \zeta_2 = 2) \quad . \quad (48)$$

For the special case of just one of the molecules being diatomic (molecule 2, for instance), then the distribution for  $x$  can be sampled directly via

$$x = \beta^{(2/\zeta_1)} \quad , \quad (\zeta_2 = 2) \quad , \quad (49)$$

with the obvious reciprocal relation applying for  $\zeta_1 = 2$ . The SSI codes recognize these special cases so that the sampling can be expedited when possible, while retaining the full generality of Eqs. (37) and (44) when required.

#### 4.3 Collisions for Molecules with Distinct Weighting Factors

There is an obvious problem when considering a collision between two simulated molecules with distinct weighting factors, since they represent a different number of real molecules. If  $W_U$  and  $W_L$  represent the weighting factors for the two molecules, with  $W_U$  being greater than  $W_L$ , then the collision is always counted as  $W_L$  "events". This is accomplished by always assigning post-collision velocity and energy components to the state vector of the molecule with the smaller weighting factor, but only changing the components of the molecule with the greater weighting factor some of the time. The probability that the

molecule with the greater weighting factor will have its components changed is simply  $W_L/W_U$ . Statistically, this means that for a large number of simulated collisions, each such simulated collision will average out to  $W_L$  real collisions for each species, even though their weighting factors differ. It should be noted that this does violate conservation of momentum and energy on an individual collision basis, but these quantities are conserved in the aggregate over a large number of collisions.

#### 4.4 Reactive Collisions

A realistic simulation of chemical reactions is a crucial element of many problems. If a bimolecular reaction of the form



has an Arrhenius rate constant of the form

$$k = A_r T^n \exp(-E_a/R_0 T) \quad , \quad (51)$$

then it is possible to define a reactive cross section as a function of translational collision energy such that the above rate constant is implied for a gas in translational equilibrium. (In Eq. (51),  $T$  is temperature,  $E_a$  is the activation energy,  $n$  is a dimensionless exponent and  $A_r$  is a prefactor.) The product of reactive cross section,  $\sigma^*$ , times relative velocity can be expressed<sup>(3)</sup>

$$v_r \sigma^* = \frac{(1 + \delta_{ij}) \sqrt{\pi} A_r}{2R_0^n \Gamma(n + 3/2)} \sqrt{1 - \frac{E_a}{E_c}} (E_c - E_a)^n \quad , \quad (52)$$

where  $\delta_{ij}$  is unity for like reactants and zero for unlike reactants. The probability of a collision resulting in a reaction is given by the ratio of the reactive to the gas kinetic cross section at the existing relative velocity between the molecules.

In the SSI Monte Carlo codes, the following procedure is used to determine which, if any, reaction will occur:

- All possible reactions are identified for the pair of molecules which are selected to experience a collision. If there are no such reactions, the procedure is bypassed.
- The probability of each allowed reaction is calculated by computing the reactive to gas kinetic cross section ratio.
- If somehow the reactive cross section(s) total to a greater value than the gas kinetic cross section (which will rarely be the case) then the probabilities are normalized by their sum.
- The collision is taken to be reactive with a probability equal to the sum of the individual reaction probabilities.
- If the collision is reactive, the reaction that occurs is selected in accordance with its probability.
- If no reaction is decided upon, then the collision is either elastic or inelastic according to the user specified probability.

Equation (52) becomes singular for  $n < -3/2$ , and this procedure cannot handle that case. The major limitation of this procedure is that it ignores the effect of internal energy in determining whether or not a reaction can occur, assuming that the entire activation energy barrier must be overcome through translational energy. If a specified fraction of the molecular internal energy is allowed to contribute to overcoming this

barrier, then the restriction on  $n$  is relaxed somewhat (see Ref. 3), but this feature is not presently implemented in the SSI codes.

#### 4.5 Post-Reaction State Definition

The post-reaction state is simply determined by adding the heat of reaction to the total collision energy (translational plus internal) and then computing reactant state vectors for the products based on this total energy and the reactants center-of-mass velocity, as for inelastic collisions. Although this is clearly an oversimplification, it is quite consistent with the phenomenological nature of the model.

The position state vector components for the products are randomly selected from the position state vector components of the reactants, which are not the same (see Section 6). The major additional complication of chemical reactions is that of distinct weighting factors. Since the reaction is  $W_L$  "events" (see Section 4.3), it only destroys the reactant with the greater weighting factor with a probability of  $W_L/W_U$ . If  $W_p$  is a product weighting factor, it is necessary to produce  $W_L/W_p$  simulated molecules of that product. In general this is not an integer quantity, so it is necessary to interpret the ratio statistically so that the expectation value is proper. That is, sometimes the next lower integer is selected and sometimes the next higher one, with a probability that reflects how close each integer value is to the desired fractional quantity. (See the discussion of molecular cloning in Section 5.2.) Of course, the weighting factors for the two products will in general be different, resulting in a different number of simulated molecules being produced for the two products. Sometimes this process could result

in a very large production of simulated molecules for a product with a small weighting factor. In order to prevent overflow of code dimensions it is necessary for the code to sense when this is happening and automatically increase the product weighting factor to prevent it.

#### 4.6 Dissociative Reactions

The situation for dissociative reactions is somewhat complicated by the presence of three rather than two products. With a little manipulation, however, it is possible to use the previous relations for this case as well. Let  $M_1$ ,  $M_2$  and  $M_3$  represent the three product molecules from a dissociative reaction, with a known center-of-mass velocity and total energy. The procedure for defining the post-reaction state is to first define an artificial complex comprised of the  $(M_2, M_3)$  pair. (There is no implication that there actually is any such collision complex.) The complex is assigned a number of internal degrees of freedom equal to  $\zeta_c$ , given by

$$\zeta_c = \zeta_2 + \zeta_3 + 2(2 - \omega) \quad , \quad (53)$$

where the last term represents the contribution of the relative translation between  $M_2$  and  $M_3$ .

The fact that this term is not simply three, as it would seem it should be, merits some explanation. It is due to the fact that these molecules are not random samples from the gas but rather special molecules owing to their being created in a reaction. This point can perhaps best be seen by considering microscopic reversibility, where the inverse reaction is a three body recombination. For this reverse process, molecules participating in it are not all equally probable, since those with greater

relative velocities are more likely to collide. Hence, the term does take on the value three for the special case of  $\omega$  equal to  $1/2$ , which is precisely the case of collision frequency being independent of relative velocity. Examination of Eqs. (37) and (44) demonstrates that translational energy in collisions behaves like another source of internal energy with  $2(2 - \omega)$  degrees of freedom.

With the number of degrees of freedom defined, the separation of  $M_1$  from the  $(M_2, M_3)$  complex is treated as an inelastic collision. The resulting velocity for the complex is then treated as the center-of-mass velocity for  $M_2$  and  $M_3$ , and the internal energy assigned to the complex becomes the total energy for the pair. Using these values,  $M_2$  and  $M_3$  are then separated in another application of the rules for inelastic collisions.

## 5. MOLECULAR TRANSLATIONS

### 5.1 Translation in an Axisymmetric Coordinate System

As discussed in Section 1, an essential element of the direct simulation Monte Carlo method is the periodic advancement of simulated molecules along their trajectories. Formally, this is accomplished by updating the position and velocity elements of the state vector. For a cartesian coordinate system this is a trivial process, but the relations are slightly more complicated for the often used axisymmetric coordinate system. Let  $v_{r0}$ ,  $v_{\phi0}$ , and  $v_{z0}$  represent the initial radial, azimuthal and axial velocity components of a molecule, with  $r_0$  and  $z_0$  representing its initial radial and axial position. Additionally, let  $\phi_0$  represent the initial azimuthal angle for the molecule. This is included here just for demonstration purposes; it will not generally be known, nor, as will be shown, will it be needed.

The initial position and velocity of the molecule can then be referenced to a standard cartesian coordinate system, yielding

$$\begin{aligned}\vec{v}_0 &= \left[ v_{r0} \cos(\phi_0) - v_{\phi0} \sin(\phi_0) \right] \vec{i} \\ &\quad + \left[ v_{r0} \sin(\phi_0) + v_{\phi0} \cos(\phi_0) \right] \vec{j} \\ &\quad + v_{z0} \vec{k} \\ &= v_{x0} \vec{i} + v_{y0} \vec{j} + v_{z0} \vec{k}\end{aligned}\tag{54}$$

and

$$\begin{aligned}\vec{r}_0 &= \left[ r_0 \cos(\phi_0) \right] \vec{i} + \left[ r_0 \sin(\phi_0) \right] \vec{j} + z_0 \vec{k} \\ &= x_0 \vec{i} + y_0 \vec{j} + z_0 \vec{k} \quad .\end{aligned}\tag{55}$$

After a time increment  $Dt$ , the position vector of the molecule will be

$$\begin{aligned}\vec{r}_1 &= (x_0 + v_{x0} Dt) \vec{i} + (y_0 + v_{y0} Dt) \vec{j} + (z_0 + v_{z0} Dt) \vec{k} \\ &= x_1 \vec{i} + y_1 \vec{j} + z_1 \vec{k} \quad ,\end{aligned}\tag{56}$$

and, in the absence of perturbing forces, the velocity vector will remain unchanged in the cartesian coordinate system. It will change in the axisymmetric coordinate system since the basis vectors are a function of position and the molecule has moved. The new radial position is

$$\begin{aligned}r_1 &= \sqrt{x_1^2 + y_1^2} \\ &= \sqrt{(r_0 + v_{r0} Dt)^2 + (v_{\phi 0} Dt)^2} \quad ,\end{aligned}\tag{57}$$

and the new axial position is

$$z_1 = z_0 + v_{z0} Dt \quad .\tag{58}$$

The radial velocity component in the coordinate system appropriate to the new molecular position can be determined by noting

$$\begin{aligned}v_{r1} &= (\vec{v}_0) \cdot (\vec{r}_1) / r_1 \\ &= \left[ v_{r0} (r_0 + v_{r0} Dt) + v_{\phi 0} (v_{\phi 0} Dt) \right] / r_1 \quad .\end{aligned}\tag{59}$$



and it can similarly be shown that

$$v_{\phi 1} = v_{\phi 0} (r_0/r_1) \quad (60)$$

and

$$v_{z1} = v_{z0} \quad . \quad (61)$$

Equations (57) - (61) give the updated position and velocity elements of the state vector for a translation in an axisymmetric coordinate system. Note that these relations are indeed independent of  $\phi_0$ . A similar procedure applies for molecular translation in other coordinate systems.

## 5.2 Molecular Cloning

When a simulated molecule is translated from one cell to another, the weighting factor for that species will generally be different in the new cell. Since it is the number of real molecules rather than the number of simulated molecules which must be preserved when crossing cell boundaries (statistically, at least), it is necessary to correct for the distinct weighting factors (see Section 3.4).

If the weighting factor before translation is  $W_0$ , then the simulated molecule represents that many real molecules. If the weighting factor in the new cell is  $W_1$ , then  $W_0/W_1$  simulated molecules would be required to represent the same number of real molecules in the new cell. If this ratio were a whole integer, then this could be accomplished by introducing that many "clones" of the molecule in the new cell. That is, that many simulated molecules would be placed in the new cell, all with the same state vector.

When the number  $W_0/W_1$  is not an integer (the usual case, of course), then the cloning must be done on a statistical basis. So, for instance, if  $W_0/W_1$  were equal to 2.7, then 30% of the time two clones would be produced and 70% of the time three would be produced. Note that the ratio may be less than unity, and the molecule may not be introduced into the new cell at all. (In which case the molecule is removed from the simulation.)

Cloning is a necessary evil inherent in a system with spatially dependent weighting factors. It enables such a system to maintain the statistically correct flux of mass and momentum across cell boundaries, but it misrepresents the flux of randomized or thermal energy. This can be seen by an extreme case where a very large number of clones is produced when a simulated molecule crosses a cell boundary. The resulting molecules in the new cell have the correct mass and momentum flux, but since they all have precisely the same velocity they have a null relative velocity, and therefore a zero temperature. If the weighting factors are not too different between adjacent cells, then the errors introduced by this process are acceptably small. However, it does mean that one cannot arbitrarily improve statistics in one portion of the solution region by selectively reducing the weighting factors there. This was a difficulty which was encountered in the early stages of the direct simulation Monte Carlo method while trying to improve statistics along the axis of axisymmetric simulations, since the cell volumes (and therefore the sample sizes) tend to be smallest on the axis.

As was the case for simulated molecules produced via chemical reactions, it is possible for the weighting factors between successive cells to be so different that a prohibitively large number of simulated molecules would be required to produce the same

number of real molecules. This is most frequently the case when a new species is being introduced, since before the species gets there the weighting factor is initialized to a very small number. The codes sense when a disproportionate number of simulated molecules are being produced for a given species and cell, and adjust the weighting factor automatically. As the weighting factor is increased, a proportionate fraction of molecules of that species and cell are removed from the simulation in order to keep the number of real molecules properly represented. This process enables the weighting factors to seek their own proper level without a priori knowledge of the solution. (Periodically, the cells are examined to determine if a certain species has been underrepresented in terms of number of simulated molecules. If it is found to be the case, then the weighting factor is decreased, allowing weighting factors to float downwards as well as upwards. It is the danger of weighting factors being too small, causing an overflow of code dimensions, which is most critical, however.)

## 6. COLLISION SAMPLING IN A MULTI-COMPONENT VHS GAS

### 6.1 General Considerations and Approach

The two general considerations in the sampling of collisions are, as usual, accuracy and efficiency of the simulation. As far as accuracy is concerned, it is crucial that the method in which molecules are selected for collisions be proper. It is imperative that the correct collision frequencies be simulated between various species, and, in fact, between the different portions of the velocity phase space for the various species. Furthermore, this frequency of simulated collisions must remain correct without any requirements put on the velocity distribution function; it certainly must not be assumed that there is a Maxwellian velocity distribution.

As far as efficiency is concerned, it is highly desirable to use a method of collision sampling involving a computational effort which is proportional to the number of simulated molecules,  $N$ , in a cell. Methods which are proportional to a power of  $N$  greater than unity can become prohibitively time consuming as the number of molecules is increased – a limit which should be made as accessible as possible for obvious physical reasons.

### 6.2 Collision Sampling for a Single Component Gas

The simplified situation of a simulation involving only one species is considered here. This problem is significant in part due to all the attention it has received and, as will be seen, it serves as an important reference case. When there is just one species, then there is just one gas kinetic cross section

(though it is still, of course, a function of collision energy), just one molecular weight and just one weighting factor for each cell. In short, just one of everything that has a molecular subscript. Hence, in this subsection all such quantities will be presented without subscripts. The most important simplification of having a single species is that there is just one collision class, i.e., only self-collisions of the given species with itself are possible.

#### 6.2.1 Collision Pair Selection

As discussed in Section 1, in the direct simulation Monte Carlo method collisions are sampled on a cell by cell basis until the number of collisions simulated is appropriate to the overall solution time step,  $\Delta t_m$ . The only spatial requirement placed on potential collision partners is that they be within the same cell. In particular, it is not required that they be within a molecular diameter of each other. (Note that if all pairs of molecules were inspected to find those that were sufficiently close to each other, this would involve a computational effort in proportion to the square of the number of molecules in the cell.) The rationale for this is that the cells should be taken small enough such that macroscopic properties can be assumed constant across the cell. When this is the case, then a molecule within the cell can be considered typical of a molecule which might exist anywhere within the cell, and molecular location can be ignored when selecting potential collision pairs.

Spatial consideration aside, the probability of any two molecules experiencing a collision is proportional to  $\sigma c_r$ , the product of their mutual cross section times their relative velocity. This probability is correctly simulated via an application

of the acceptance-rejection method if pairs of molecules are selected at random, and then kept or rejected as collision partners with a probability proportional  $\sigma c_r$ . This is accomplished by keeping a maximum value for  $\sigma c_r$  for each cell (which is updated if a larger value is encountered) and computing the ratio  $r$  defined by

$$r = \frac{\sigma c_r}{(\sigma c_r)_{\max}} \quad . \quad (62)$$

A random variable,  $\beta$ , is then generated, and the pair of molecules is accepted as collision partners if  $r$  is greater than  $\beta$ . This produces the proper relative collision probability without regard to the existing velocity distribution function.

#### 6.2.2 Collision Time Counter for a Single Component Gas

The volumetric collision frequency for a single component gas,  $\nu$  (collisions per unit volume per unit time), is given by

$$\nu = \frac{1}{2} n^2 \overline{\sigma c_r} \quad , \quad (63)$$

where, as in Section 2,  $n$  represents the number density of the species and  $\overline{\sigma c_r}$  is the average product of collision cross section and relative velocity. At first inspection, it would seem from Eq. (63) that a correct simulation of collision frequency would require evaluation of  $\overline{\sigma c_r}$ , which would mean that all pairs of molecules in a cell would have to be considered. Such a procedure involves a computational effort proportional to  $N^2$  and is to be avoided, if possible, in preference to a method which is simply proportional to  $N$ .

The alternative approach, introduced by Bird, is the time counter approach. For each collision a time counter,  $t_c$ , is incremented by an amount that depends on the relative velocity of the collision. Collision sampling continues in a cell until its time counter has been advanced beyond the overall flow simulation time, at which time the code proceeds to the next cell (which has its own time counter). The time counter increment,  $\Delta t_c$  is given by

$$\Delta t_c = \frac{2}{Vn^2\sigma c_r}, \quad (64)$$

where  $V$  is the cell volume and  $n$  is the species number density given by

$$n = NW/V, \quad (65)$$

with  $W$  being the weighting factor for the species. (Equation (65) is just a special case of Eq. (26).) It should be stressed that Eq. (64) applies for each real collision. As is discussed in Sections 3.4 and 4.3, each simulated collision corresponds to  $W$  real collisions, so when a simulated collision occurs the actual applied increment to  $t_c$  is  $W$  times the value given by Eq. (64).

It is not obvious that Eq. (64) will lead to a proper simulation of the overall collision frequency, so a demonstration will be presented. Let  $f_1(c_r)$  be the normalized distribution function for relative velocity appropriate to a given cell at a given time. (Note that most problems solved by a Monte Carlo technique involve repeated runs, where the total number of collision pairs can be made arbitrarily large. The introduction of a distribution function, and therefore the demonstration of the correctness of Eq. (64) is strictly valid only in the limit of infinitely many runs.

Since the essence of the method is a correct statistical representation, it could not be otherwise.) By definition,

$$\int_0^{\infty} f_1(c_r) dc_r = 1 , \quad (66)$$

irrespective of the particular form for  $f_1$ . Let  $f_2(c_r)$  be the normalized distribution function of relative velocity occurring in collisions. Since collisions occur with a probability proportional to  $\sigma c_r$ ,  $f_2$  can be constructed from  $f_1$  via

$$f_2(c_r) = f_1(c_r) \frac{\sigma c_r}{\overline{\sigma c_r}} , \quad (67)$$

where  $\overline{\sigma c_r}$  can now be formally defined via

$$\overline{\sigma c_r} = \int_0^{\infty} f_1(c_r) \sigma c_r dc_r . \quad (68)$$

(Note that in Eq. (68), as in the rest of this demonstration, the functional form of the dependence of cross section on relative velocity need never be specified. The time counter represented by Eq. (64) is therefore not restricted to any particular model for the cross section.)

The average increment of the time counter that is applied over many simulated collisions is therefore given by

$$\overline{\Delta t_c} = \int_0^{\infty} f_2(c_r) \Delta t_c(c_r) dc_r . \quad (69)$$

If Eqs. (64) and (67) are substituted into Eq. (69), the result is

$$\overline{\Delta t_c} = \frac{2}{n^2 V \overline{\sigma c_r}} , \quad (70)$$



where the normalization condition (Eq. (66)) has been utilized. The implication of Eq. (70) is that, on average, the frequency of collisions in the cell is  $Vn^2\overline{\sigma c_r}/2$ . Since this is simply the product of cell volume times the proper volumetric collision frequency (Eq. (63)), the validity of the time counter given in Eq. (64) has been demonstrated.

### 6.3 Collision Class Sampling in Gas Mixtures

The above procedure for a single species gas can be extended to a multi-component mixture via consideration of distinct collision classes. In this approach, collision classes are defined by the colliding pair identities. Hence, if there are  $p$  species in the simulation then there are  $p(p+1)/2$  collision classes, which can be identified by the subscripts of the corresponding molecular pair. (The number of classes is not  $p^2$  since the order of molecule specification is not taken to matter in determining a collision class. Hence, the class identified by the subscripts  $i,j$  is not distinct from the class identified by the subscripts,  $j,i$ .)

In collision class sampling, which is the method used by Bird,<sup>(1)</sup> each collision class is sampled separately, and the collision sampling in a cell is not complete until all classes have been considered. Each collision class has its own stored value of  $(\sigma_{ij}c_r)_{\max}$ , and its own separate time counter,  $t_{cij}$ . By a comparable analysis to that presented above, it can be shown that the appropriate time counter increment in this case is

$$\Delta t_{cij} = \frac{(1 + \delta_{ij})}{n_i n_j V \sigma_{ij} c_r} \quad , \quad (71)$$

where, as in Section 4,  $\delta_{ij}$  is the Kronecker delta which is unity for  $i = j$  and zero otherwise. As in the previous section,

the above increment applies for each real collision. A simulated collision corresponds to  $W_L$  real collisions, where  $W_L$  is the lesser of  $W_i$  and  $W_j$  (see Section 4.3), so when a simulated collision occurs, the applied increment to  $t_{cij}$  is  $W_L$  times the result of Eq. (71).

#### 6.4 Global Collision Sampling in a Gas Mixture

Although the procedure described above is quite reasonable for, say, a two component mixture, it becomes quite complicated as the number of species increases. For 10 species, for instance, the program must loop over 55 distinct collision classes for each cell, and storage must be allocated for 110 quantities in each cell. As the number of species increases, the storage requirement for the collision sampling constants quickly becomes greater than the storage required for the molecular state vectors! The obvious simplification is to search for a technique where collisions are simulated simultaneously for all collision classes, with each class having its proper relative probability of being selected. The overall collision sampling then continues until a single time counter indicates that sufficient collisions have been sampled in the current time step and cell.

##### 6.4.1 Global Collision Time Counter

If molecular pairs are selected for collisions such that the various collision classes automatically appear with the proper relative frequency (see below), then it is not necessary to consider separate time counters for all the various collision classes. One approach that could then be applied is to keep a collision time counter for just one collision class, and increment it when collisions of that class occur. If the various collision classes are being selected according to their correct

relative frequency, then simulating the proper frequency for one collision class will ensure, in the long run, that all collision classes are occurring with the correct frequency. A disadvantage with this approach is the necessity of making an arbitrary choice for the collision class which is to have a time counter. Furthermore, there may be no good choice in a reacting flow where the dominant species may vary strongly from place to place. (Clearly, one would not want to select a class of collision that does not occur in a given cell, since the result would be a never ending sampling of collisions of other classes.)

The preferred approach is to define a global collision time counter,  $t_g$ , which is a weighted average of the time counters of all collision classes; i.e.

$$t_g = \frac{\sum_{i=1}^p \sum_{j=1}^i D_{ij} t_{cij}}{\sum_{i=1}^p \sum_{j=1}^i D_{ij}} , \quad (72)$$

where the  $D_{ij}$  are nonnegative coefficients which can be selected at will. Note that in this formulation every collision will result in some increment of the global time counter (unless  $D_{ij} = 0$  for that class), so the collision sampling frequency is not dependent on any one collision class.

It remains, of course, to specify the  $D_{ij}$ . A very convenient choice is given by

$$D_{ij} = \frac{n_i n_j}{(1 + \delta_{ij})} . \quad (73)$$

Firstly, Eq. (73) is convenient because it tends to make the collision classes with the higher collision frequencies count more, resulting in good statistics for  $t_g$  irrespective of cell location. (Note that  $D_{ij}$  is cell dependent since the species number densities are cell dependent.) Secondly, Eq. (73) results in a particularly convenient form for  $t_g$ . The normalizing factor in the denominator of Eq. (72) can be summed to give

$$t_g = \frac{2}{n^2} \sum_{i=1}^p \sum_{j=1}^i \frac{n_i n_j}{(1 + \delta_{ij})} t_{cij} \quad . \quad (74)$$

Hence, a collision of class  $ij$ , which would produce an increment of  $\Delta t_{cij}$  to its own time counter produces an increment  $\Delta t_g$  to  $t_g$  given by

$$\Delta t_g = \frac{2}{n^2} \left( \frac{n_i n_j}{1 + \delta_{ij}} \right) \Delta t_{cij} \quad , \quad (75)$$

where, again,  $n$  is the total number density of all species in the cell. If Eq. (71) is substituted into Eq. (75), the result is

$$\Delta t_g = \frac{2}{V n^2 \sigma_{ij} c_r} \quad . \quad (76)$$

Equation (76) is extremely significant since it recaptures the precise form of the time counter increment for a single species (Eq. (64)), but indicates that it is completely valid for a multi-component mixture so long as the various collision classes are sampled with the proper relative frequency.

#### 6.4.2 Collision Pair Selection in Multi-Component Mixtures

When considering selection of collision pairs, it is crucial to remember the distinction between real and simulated molecules discussed in Section 3.4. Given two simulated molecules selected at random from within the cell, the probability of their having a real collision is proportional to  $W_i W_j \sigma_{ij} c_r$ . However, real collisions cannot happen individually; they come  $W_L$  at a time, where  $W_L$  is the lesser of  $W_i$  and  $W_j$ . Hence, when a collision is decided upon in the program,  $W_L$  of them will occur. To compensate for this, potential collision pairs should be accepted for collision according to the size of  $Q$  given by

$$Q = W_U \sigma_{ij} c_r \quad . \quad (77)$$

The relative frequency of real  $ij$  collisions will then be proportional to the product  $QW_L$  (the relative probability of a pair being accepted for collision times the number of real collisions occurring when the pair is accepted), which is the desired relation. Selection of collision pairs with the correct relative frequency then assures that incrementing the global time counter as discussed above will give a statistically correct sampling of all collision classes simultaneously.

#### 6.4.3 Summary of Collision Sampling in Multi-Component Mixtures

The results of this section can be summarized via the following procedure for the sampling of collisions:

- Each cell has a (current) maximum value of  $Q$ ,  $Q_{\max}$ , that has been encountered so far in the collision sampling process. Whenever a larger value is encountered,  $Q_{\max}$  is set equal to that larger value.

- Each cell has a current value of the global time counter,  $t_g$ .
- Pairs are selected at random from all molecules within the cell.
- For each pair,  $Q$  (as defined by Eq. (77)) is computed.
- The ratio of  $Q$  to  $Q_{\max}$  is computed, and a random variable is generated. The pair is accepted for collision if the random variable is less than the ratio. (If the pair is not accepted, then another random pair is selected. The process continues until a pair is accepted.)
- For an accepted pair, the collision mechanics are computed as described in Section 4. This always corresponds to  $W_L$  collisions.
- The global time counter is incremented by  $W_L \Delta t_g$ , where  $\Delta t_g$  is given in Eq. (76).
- The process continues until the global time counter goes beyond the overall flow time. At that point, the collision sampling is commenced in the next cell.
- When all cells have had collisions simulated, then the code proceeds to the translation portion. (See Sections 1 and 5.)

## 7. INITIAL AND BOUNDARY CONDITIONS

### 7.1 General Considerations

The initial and boundary conditions necessary to simulate a problem frequently do not receive their fair share of consideration. It is these conditions which usually distinguish one solution from another, and their correct and efficient specification should be a central concern. Nonetheless, there is clearly room for advancements, particularly in the specification of boundary conditions. Many gas dynamic solutions involve boundary conditions specified at infinity, which are currently simulated by placing boundaries very far from the main flow region. It would result in a substantial computational simplification if boundary conditions applicable closer in to the flow region of interest could be generated for free gas boundaries. Wall boundary conditions also frequently involve a fair degree of approximation, usually taking the form of accommodation coefficients (though in this case the problem is as much a lack of basic physical understanding of the gas-surface interaction process as it is a lack of a good numerical simulation).

### 7.2 Initial Conditions

Since the direct simulation Monte Carlo method is inherently an unsteady technique, an initial state must be specified in order to advance the solution. (For situations where a steady state result is desired, it is obtained as the long time solution to an unsteady problem. In this case the initial conditions have no effect on the eventual solution, but they may well have an impact on the speed with which that state is achieved.) It

will be assumed here that the initial conditions correspond to a uniform flow with the translational and internal modes being in equilibrium. The specification of the initial conditions therefore involves determining the state vector elements consistent with this condition for the desired number of molecules.

#### 7.2.1 Number of Simulated Molecules and Weighting Factors

The desired number of simulated molecules of each species in each cell (referred to here as  $M_c$ ) will usually be an input quantity. (Typically, simulations aim for a total number of molecules per cell in the neighborhood of twenty.) Given the initial number density to be simulated for a species,  $n_i$ , (which will have been converted in the code to internal dimensions - see Section 3) the weighting factor for a species in a cell can be derived from an application of Eq. (26) to give

$$W_i = Vn_i/M_c \quad , \quad (78)$$

where  $V$  is the cell volume. If a species is not initially present in a cell, then the weighting factor is set to a very small but positive number, typically that which would correspond to an initial mole fraction of one part per million or so. A weighting factor of zero would cause an attempt to divide by zero when molecules of that species move into the cell, but it is generally best to start the weighting factors off small. As the solution proceeds, the weighting factors are automatically adjusted, but the adjustment upward is more direct and immediate than the adjustment downward (see Sections 3.4 and 5.2).



### 7.2.2 Initial Positions

The initial molecules assigned to a cell should have an equal probability of being placed in any volume element of the cell. The rules for accomplishing this will change with the coordinate system being used, but they are readily derivable from the general principle. As an example, consider an axisymmetric simulation where a cell is bounded by the radial positions  $r_1$  and  $r_2$  ( $r_1 < r_2$ ), and the axial coordinates  $z_1$  and  $z_2$  ( $z_1 < z_2$ ). Since the basic volume element in this coordinate system is  $2\pi r dr dz = \pi d(r^2) dz$ , the volume elements will be sampled with equal probability if  $r^2$  and  $z$  are sampled randomly. Hence, a molecule can be assigned axial and radial positions via

$$r = \sqrt{r_1^2 + \beta(r_2^2 - r_1^2)} \quad (79)$$

and

$$z = z_1 + \beta(z_2 - z_1) \quad (80)$$

(Recall that every time the symbol  $\beta$  occurs it implies a separate random variable. In particular, one should certainly not use the same random variable to determine radial and axial coordinates. The second application would hardly qualify as "random".)

### 7.2.3 Initial Velocity Components

The thermal velocity components for a molecule in translational equilibrium (neglecting, for the moment, any mean flow contribution) should be selected from the normalized Maxwellian velocity distribution,  $f_0(v)$ , given by

$$f_0 = \frac{\alpha}{\sqrt{\pi}} \exp [-(\alpha v)^2] \quad , \quad (81)$$

where

$$\alpha = \sqrt{\frac{m}{2R_0T_\infty}} \quad , \quad (82)$$

$m$  is the species molecular weight,  $R_0$  is the universal gas constant and  $T_\infty$  is the temperature. Equation (81) applies for each of the molecular velocity components, and must be sampled three times for each molecule that comprises the initial state of the simulation. A method for directly sampling from this distribution, as presented in Ref. 1, is

$$A_1 = 2\pi\beta \quad , \quad (83)$$

$$A_2 = \sqrt{-\log(\beta)}/\alpha \quad , \quad (84)$$

$$v = A_2 \sin(A_1) \quad . \quad (85)$$

After the thermal velocity components are determined for each molecule, then any mean flow velocity is simply added on. The velocities are then transformed to internal units (see Section 3.3).

#### 7.2.4 Initial Internal Energies

The only remaining element of the state vector to be specified is the internal energy. Internal energies for a gas in equilibrium are distributed according to the normalized distribution function  $f_I$  given by

$$f_I = \frac{\xi^{\zeta/2-1}}{\Gamma(\zeta/2)} \exp(-\xi) \quad , \quad (86)$$

where  $\zeta$  represents the number of internal degrees of freedom for the species in question,  $\Gamma$  is the gamma function and  $\xi$  is a dimensionless internal energy, i.e.

$$\xi = E_I / R_0 T_\infty , \quad (87)$$

where  $E_I$  is the internal energy. In general, sampling of Eq. (86) must be done via the acceptance-rejection method. In the present SSI codes  $\zeta$  is restricted to being greater than or equal to two (or the trivial case of  $\zeta$  equal to zero, which just gives  $E_I$  identically equal to zero). If  $\zeta$  is precisely equal to two, then a direct sampling of the internal energy is possible via

$$\xi = -\log(\beta) \quad . \quad (\zeta = 2) \quad (88)$$

In the general case of  $\zeta > 2$ , it proves convenient to first introduce the transformation  $s = \sqrt{\xi}$ .  $s$  is then distributed in proportion to the distribution  $g(s)$  given by

$$g(s) = 2s^{(\zeta-1)} \exp(-s^2) \quad . \quad (89)$$

Since  $g(s)$  is to be sampled via the acceptance-rejection method, it is first necessary to determine its maximum value,  $g_{\max}$ . Standard calculus serves to show that  $g_{\max}$  occurs at  $s = s^*$ , where

$$s^* = \sqrt{(\zeta - 1)/2} \quad , \quad (90)$$

so

$$\frac{g(s)}{g_{\max}} = \left( \frac{s}{s^*} \right)^{\zeta-1} \exp[s^{*2} - s^2] \quad . \quad (91)$$

The sampling of Eq. (89) proceeds as follows:

- A value of  $s$  is selected randomly from the interval  $s_{\min}$  to  $s_{\max}$ , where

$$s_{\max} = s^* + 5 \quad (92)$$

and

$$s_{\min} = \text{greater of } \{0, s^* - 5\} \quad . \quad (93)$$

(Note that in this interval  $g(s)$  goes from its maximum value to a value on the order of  $10^{-10}$  its maximum value. The transformation from Eq. (86) to Eq. (89) was made mainly to achieve a probability function which dies off extremely rapidly away from its maximum value, so that very little error is associated with considering a finite subinterval for the sampled variable.)

- $g(s)/g_{\max}$  is calculated via Eq. (91).
- A random variable is generated, and the value of  $s$  is kept if the random variable is less than  $g(s)/g_{\max}$ . Otherwise, the procedure is repeated until a value of  $s$  is accepted.
- When a value of  $s$  is selected, then the internal energy is given by  $s^2 R_0 T_\infty$ .

As for all the initial conditions, the codes will automatically then express the values in internal dimensions (see Section 3.3).

### 7.3 Wall Boundary Conditions

Boundary conditions in the direct simulation Monte Carlo technique are applied for both wall and free gas boundaries. In a wall boundary condition, whenever a molecule is simulated to strike the wall some action must be taken. Unless the wall is a condensing boundary the molecule will be reflected, and one of several boundary conditions can be selected. The easiest condition is that of a specularly reflecting wall, where the molecule simply is replaced by its mirror image to keep it within the solution region. Another frequently used condition is that of a perfectly accommodating wall. In this condition, a molecule

is reemitted from the wall after being selected from a distribution characteristic of the wall temperature and velocity. To date, wall boundary conditions have not been required in the SSI codes, and they will not be discussed further here. They are discussed at some length in Ref. 1, including the intermediate case of partial accommodation.

#### 7.4 Free Gas Boundary Conditions

In many cases, the boundary condition is meant to simulate a region of uniform equilibrium flow. Molecules leave the solution region in the normal course of their trajectories, and they simply disappear from the simulation. Molecules are introduced from outside the boundary as selected from distributions appropriate to incoming molecules in the undisturbed flow. It should be stressed that this is not the same as simply sampling a Maxwellian velocity distribution, since it is the molecular flux across the boundary which must be correctly simulated. Hence, molecules with a large component of velocity inward from the boundary are more likely to be selected than they would be in choosing molecules appropriate to a static distribution as was done for the initial conditions described above.

##### 7.4.1 Incoming Number Flux

The first requirement is to determine the number of molecules which should be introduced across the boundary during the solution time step  $\Delta t_m$ . The incoming number flux,  $q$ , (molecules per unit area per unit time) can be expressed<sup>(1)</sup>

$$q = n_i \left\{ \exp(-w^2)/\sqrt{\pi} + w[1 + \operatorname{erf}(w)] \right\} / 2\alpha, \quad (94)$$

where

$$w = \alpha u_{\infty} \cos(\theta) \quad , \quad (95)$$

and  $\theta$  represents the angle between the inward surface normal and the mean flow, which has a magnitude  $u_{\infty}$  (i.e.,  $u_{\infty} \cos(\theta)$  is the inward component of the mean flow velocity),  $n_i$  is the far field number density of the species of interest and  $\alpha$  is given in Eq. (82).

Equation (94) must be applied for each cell on the boundary and for each species that exists in the ambient. The number of simulated molecules to be introduced into the cell,  $N_b$ , is given by

$$N_b = q A_c \Delta t_m / W \quad , \quad (96)$$

where  $A_c$  is the area of the cell along the boundary and  $W$  is the weighting factor for the species and cell in question.

#### 7.4.2 Incoming Molecular Velocity Components

A coordinate system should be set up locally at the boundary such that one direction is in the direction of the inward normal and the other two directions are perpendicular to it. Velocity components are first determined in terms of this local coordinate system, and then transformed, if necessary, to the main code coordinate system. In the local coordinate system, the velocity components parallel to the surface are determined as discussed above for initial state molecules, but the inward component of velocity must be selected in proportion to the distribution  $h(v)$  given by

$$h(v) = (v + w) [\exp -(\alpha v)^2] \quad , \quad (97)$$

which must be sampled via the acceptance-rejection method. (Only positive  $v$  is allowed, of course, from the definition of the coordinate system.)

#### 7.4.3 Incoming Molecular Position

An initial entry position should be selected for the molecule such that the flux is randomly distributed. Assuming there is no variation of  $\theta$  across  $A_c$  this means that each area element of the exposed cell area should have an equal probability of being selected. Once the initial entry position is selected, then the molecule should be translated a random fraction of  $\Delta t_m$  along its trajectory to determine its actual location. All of the considerations discussed in Section 5 with regard to molecular translations apply to this translation and, in particular, it must be possible to dynamically adjust the weighting factor as required.

#### 7.4.4 Incoming Molecular Internal Energies

The internal energies are selected as described in Section 7.2.4.

## 8. STATISTICAL SAMPLING OF OUTPUT

### 8.1 General Considerations

It is safe to say that the molecular state vectors as they exist in the computer do not comprise the desired output of the procedure. With rare exceptions, it is usually macroscopic quantities such as temperature, density, mean flow velocity, etc., which are of interest - not the microscopic quantities represented by the state vector components of an individual simulated molecule. The generation of the desired output requires that the macroscopic quantities of interest be represented in terms of statistical sums of the available microscopic quantities; and it is the main purpose of this section to present these correspondences. All sums are kept in terms of "real" molecules and events, i.e., the current weighting factors are included in the sums. This is essential since the weighting factor determines the statistical importance of a given molecule. Since the weighting factors are dynamically and unpredictably adjusted as the solution progresses, it would not be possible to go back and add in the effect of weighting factors a posteriori.

In general, it must be decided ahead of time exactly what output is desired from the code, and therefore what statistical sums should be kept to generate it. There is a vast amount of potential information in the simulation, and it is not reasonable to store all possibly interesting quantities in all runs. On the other hand, it is wasteful to completely rerun a case just because the user decides there was an additional quantity he was interested in. The selection of output for a given run, therefore, unavoidably requires user judgement. Once the



user has decided upon the required output, the determination of which statistical sums should be kept can be done automatically by the code. Care is taken to make sure that a statistical sum is not duplicated internally if it is required by more than one requested output quantity.

Some initial words of caution are required. By its nature, the direct simulation Monte Carlo method works with far fewer molecules than nature does, and it therefore exhibits considerably greater statistical variation in its macroscopic predictions. To reduce these variations, the codes are run repeatedly for the same case, increasing the statistical base from which the macroscopic output is derived. If care is taken to use efficient techniques, such as described in this report, then useful results can usually be obtained with a modest computational effort. This statement must be tempered, however, by a realization of the convergence rate for Monte Carlo sampling. Basically, the statistical error in the output converges as one over the square root of the sample size (or run time). Hence, if a solution looks good, but the user decides he would like one more significant digit (i.e., he would like the statistical error to be reduced to 0.1 times its current value) it would require that the run time be increased by a factor of 100! It can be seen that the desire for more accuracy can very quickly turn the most efficient code into a money gobbling nightmare. When using a Monte Carlo technique, one must accept some statistical scatter in the output.

## 8.2 Sampling of Instantaneous Output Quantities

Instantaneous output quantities are those which are, in principle, derivable from an instantaneous "snapshot" of the solution. These quantities, such as density, temperature and velocity, can be determined by examining the molecular state

vectors at a particular time in the simulation. The code pauses in the simulation and uses the molecular state vector elements to add values to statistical sums appropriate to the various cells and the particular time that it paused. It then proceeds with the simulation until the next sampling time. As the code goes through its successive runs, it stops at the same points in the simulation every time and adds to the statistical base for the sums. The items listed below, with their statistical definitions, are selectable as output requests in the SSI codes. Summations are made over all applicable simulated molecules, which includes  $N_{\text{run}}$  separate runs.

Total Number Density

$$n = \frac{1}{VN_{\text{run}}} \sum_i W_i \quad (98)$$

Mean Molecular Weight

$$m = \frac{\sum_i W_i m_i}{\sum_i W_i} \quad (99)$$

j'th Velocity Component

$$V_j = \frac{\sum_i W_i m_i V_{ji}}{\sum_i W_i m_i} \quad (100)$$

Overall Translational Temperature

$$T = \frac{1}{3R_0 S_1} \left( S_2 - \frac{S_3^2 + S_4^2 + S_5^2}{S_6} \right) \quad (101)$$

where

$$S_1 = \sum_i W_i \quad (102)$$

$$S_2 = \sum_i W_i m_i (v_{1i}^2 + v_{2i}^2 + v_{3i}^2) \quad (103)$$

$$S_3 = \sum_i W_i m_i v_{1i} \quad (104)$$

$$S_4 = \sum_i W_i m_i v_{2i} \quad (105)$$

$$S_5 = \sum_i W_i m_i v_{3i} \quad (106)$$

$$S_6 = \sum_i W_i m_i \quad (107)$$

Translational Temperature in j'th Direction

$$T_j = \frac{1}{R_0 S_1} \left( S_7 - S_8^2 / S_6 \right) \quad (108)$$

where

$$S_7 = \sum_i w_i m_i v_{ji}^2 \quad (109)$$

$$S_8 = \sum_i w_i m_i v_{ji} \quad (110)$$

Internal Mode Temperature

$$T_I = \frac{2 \sum_i w_i E_{Ti}}{R_0 \sum_i w_i \zeta_i} \quad (111)$$

With the exception of Eq. (99), all of the above quantities can also be defined and calculated for any specified species. The sums are the same except that only molecules of that species are considered. Before printing output quantities, they are always transformed to standard dimensions from the internal scales.

### 8.3 Sampling of Time Averaged Quantities

Some additional quantities of interest are not sampled at a separate sampling time as described above, but rather as the simulation evolves. Examples of such quantities are collision rates, reaction rates, mean velocities between molecules, etc. In general, these quantities depend on the relative state of more than one type of molecule, and they are by their nature

expressed as average values over a finite time interval. The formulas for calculating these quantities are little more than event counters, and will not be included here. The following quantities are currently available as output:

- Mean Relative Velocity Between any Two Species
- R.M.S. Deviation of Mean Relative Velocity Between any Two Species
- Mean Product of Cross Section Times Relative Velocity Between any Two Species
- Collision Rate Between any Two Species
- Reaction Rate for any Chemical Reaction.

The sampling for all of these quantities occurs in the collision simulation routines. As pairs are considered as possible collision partners, statistics are kept, if necessary, to generate the first three quantities. Statistics on collisions and reactions are kept as they occur.

## 9. REFERENCES

1. Bird, G.A., Molecular Gas Dynamics, Clarendon Press, Oxford, 1976.
2. Chapman, Sydney, and Cowling, T.G., The Mathematical Theory of Non-Uniform Gases, 3rd ed., Cambridge University Press, Cambridge, 1970, pp. 86-87.
3. Bird, G.A., "Monte-Carlo Simulation in an Engineering Context," Proceedings of the 12th International Symposium on Rarefied Gas Dynamics, Vol. 74, Progress in Astronautics and Aeronautics, AIAA, New York, 1981.
4. Borgnakke, Claus, and Larsen, Paul S., "Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture," Journal of Computational Physics, Vol. 18, 1975, pp. 405-420.
5. Vincenti, Walter G., and Kruger, Charles H., Jr., Introduction to Physical Gas Dynamics, John Wiley and Sons, 1965, pp. 348-356.